

```

1 // [團體佇列/Queue](1/3)
2 #define IN "P18IN.txt"
3 #define OUT "P18OUT.txt"
4 //*****
5 #include <iostream>
6 #include <ctime>
7 using namespace std;
8 void redir(void);
9 //*****
10 /* Work Space*/
11 #include <map>
12 #include <queue>
13 #include <string>
14 //*****
15 int main(void)
16 {
17     redir(); //redirection
18 //*****
19 /* Work Space*/
20     int t, n, x;
21     int i;
22     int t0;
23     int kase = 0;
24
25     while(scanf("%d", &t) == 1 && t){ //t: the number of teams, 0 for quit
26         map<int, int> team; //block scope
27         for(i=0; i<t; i++){
28             scanf("%d", &n); //n: the number of elements of team i
29             while(n--){
30                 scanf("%d", &x); //x: element
31                 team[x] = i;
32             }
33         }
34
35         queue<int> teamQueue;
36         queue<int> elementQueue[1000]; //elementQueue[i]: queue of elements of team i
37
38         printf("Scenario #%d\n", ++kase);
39         while(1){
40             string cmd; //block scope
41             cin >> cmd;
42
43             if(cmd[0] == 'S'){ //STOP
44                 break;
45             } else if(cmd[0] == 'D'){ //DEQUEUE
46                 t0 = teamQueue.front();
47                 cout << elementQueue[t0].front() << endl;
48                 elementQueue[t0].pop();
49                 if(elementQueue[t0].empty()){
50                     teamQueue.pop();
51                 }
52             } else if(cmd[0] == 'E'){ //ENQUEUE
53                 scanf("%d", &x);
54                 t0 = team[x];
55                 if(elementQueue[t0].empty()){
56                     teamQueue.push(t0);
57                 }
58                 elementQueue[t0].push(x);
59             }
60         }
61         printf("\n");
62     }
63 //*****

```

```

64 //[團體佇列/Queue](2/3)
65     freopen("CON", "r", stdin); //取消重新導向
66     freopen("CON", "w", stdout);
67
68     printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
69
70     system("pause");
71     return 0; //the end...
72 }
73
74 void redir(void)
75 {
76     freopen(IN, "r", stdin);
77     freopen(OUT, "w", stdout);
78 }
79 //*****
80 /* Work Space*/
81 //Input(IN) Sample
82 /*
83 2
84 3 101 102 103
85 3 201 202 203
86 ENQUEUE 101
87 ENQUEUE 201
88 ENQUEUE 102
89 ENQUEUE 202
90 ENQUEUE 103
91 ENQUEUE 203
92 DEQUEUE
93 DEQUEUE
94 DEQUEUE
95 DEQUEUE
96 DEQUEUE
97 DEQUEUE
98 STOP
99 2
100 5 259001 259002 259003 259004 259005
101 6 260001 260002 260003 260004 260005 260006
102 ENQUEUE 259001
103 ENQUEUE 260001
104 ENQUEUE 259002
105 ENQUEUE 259003
106 ENQUEUE 259004
107 ENQUEUE 259005
108 DEQUEUE
109 DEQUEUE
110 ENQUEUE 260002
111 ENQUEUE 260003
112 DEQUEUE
113 DEQUEUE
114 DEQUEUE
115 DEQUEUE
116 STOP
117 0
118 */
119
120
121
122
123
124
125
126

```

```
127 //[團體佇列/Queue](3/3)
128 //Output(OUT)
129 /*
130 Scenario #1
131 101
132 102
133 103
134 201
135 202
136 203
137
138 Scenario #2
139 259001
140 259002
141 259003
142 259004
143 259005
144 260001
145 */
```